

Sampling from a polytope and hard-disk Monte Carlo

Sebastian C. Kapfer and Werner Krauth

E-mail: sebastian.kapfer@ens.fr, werner.krauth@ens.fr

Laboratoire de Physique Statistique, Ecole Normale Supérieure, UPMC, CNRS
24 rue Lhomond, 75231 Paris Cedex 05, France

Abstract. The hard-disk problem, the statics and the dynamics of equal two-dimensional hard spheres in a periodic box, has had a profound influence on statistical and computational physics. Markov-chain Monte Carlo and molecular dynamics were first discussed for this model. Here we reformulate hard-disk Monte Carlo algorithms in terms of another classic problem, namely the sampling from a polytope. Local Markov-chain Monte Carlo, as proposed by Metropolis et al. in 1953, appears as a sequence of random walks in high-dimensional polytopes, while the moves of the more powerful event-chain algorithm correspond to molecular dynamics evolution. We determine the convergence properties of Monte Carlo methods in a special invariant polytope associated with hard-disk configurations, and the implications for convergence of hard-disk sampling. Finally, we discuss parallelization strategies for event-chain Monte Carlo and present results for a multicore implementation.

1. Introduction

The hard-disk system is a fundamental model of statistical and computational physics. During more than a century, the model and its generalization to d -dimensional spheres have been central to many advances in physics. The virial expansion is an example: Boltzmann's early calculations of the fourth virial coefficient [1] ultimately led to Lebowitz and Onsager's proof of the convergence of the virial expansion up to finite densities [2] for all d and to the general and systematic study of virial coefficients. The theory of phase transitions provides another example for the lasting influence of the hard-disk model and its generalizations. Kirkwood and Monroe [3] first hinted at the possibility of a liquid–solid transition in three-dimensional hard spheres. This prediction was surprising because of the absence of attractive interactions in this system. The depletion mechanism responsible for the effective-medium attraction was also first studied in hard spheres, by Asakura and Oosawa [4]. In two dimensions, the liquid–solid phase transition was first evidenced by Alder and Wainwright [5]. It led to far-reaching theoretical [6], computational [7, 8] and experimental [9] work towards the understanding of 2D melting. In mathematics, hard disks and hard spheres have also been at the center of attention [10]. A rigorous existence proof of the melting transition in hard spheres is still lacking, but the ergodicity of the molecular dynamics evolution of this system has now been established rigorously [11, 12].

Arguably the most important role for the hard-disk model has been in the development of numerical simulation methods. Molecular dynamics [13, 14] and Markov-chain Monte Carlo [15] were first formulated for hard disks. The early algorithms have continued to be refined: Within the molecular dynamics framework, this has led to highly efficient event-scheduling strategies [14, 16] and, for Monte Carlo, to the development of cluster algorithms [17, 18, 19].

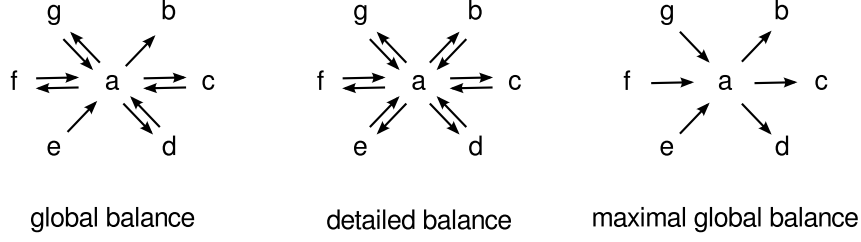


Figure 1. Monte Carlo balance conditions: Arrows represent probability flows $P_{a \rightarrow b} = \pi_a p_{a \rightarrow b}$ between configurations a and b (each arrow stands for a probability flow of same magnitude). *Left:* Global balance, as required for Markov-chain Monte Carlo. The total flow $\sum_c P_{c \rightarrow a}$ into the configuration a equals the flow $\sum_c P_{a \rightarrow c}$ out of it. *Center:* Detailed balance: the net flow between any two configurations is zero, $P_{a \rightarrow b} = P_{b \rightarrow a}$. *Right:* Another special case of global balance: maximal global balance at a ($P_{a \rightarrow b} > 0 \implies P_{b \rightarrow a} = 0$).

Even the modern simulation algorithm remain slow, however, and revolutions like the cluster algorithms for spin systems [20, 21] have failed to appear. Moreover, rigorous mathematical bounds for the correlation time (mixing time) of Monte Carlo algorithms were obtained in the thermodynamic limit only for small densities [22, 23, 24], which are far inside the liquid phase. At higher densities, close to the liquid–solid transition, many numerical calculations have suffered from insufficient simulation times until recently [7, 8].

In the present article, we discuss computational aspects of the hard-disk model, starting with an introduction (Section 2). In particular, we reinterpret hard-sphere Monte Carlo in terms of the sampling of points from high-dimensional polytopes (Section 3). Local Monte Carlo amounts to random walks in a sequence of such polytopes, while event-chain Monte Carlo is equivalent to molecular dynamics evolutions with particular initial conditions for the velocities. We analyze the convergence properties of the algorithms in these polytopes for the hard-disk case. Parallel event-chain algorithms emerge naturally as molecular dynamics with more general initial conditions (Section 4). We describe several parallelization strategies and report on implementations.

2. Local Monte Carlo and event-chain Monte Carlo

We consider N equal hard disks of unit radius $\sigma = 1$ in a square box of size $L \times L$. In the following, we assume without mentioning periodic boundary conditions for positions and pair distances. The statistical weights π_a are equal to unity for configurations a without overlaps (all pair distances larger than 2) and zero for illegal configurations (with overlaps). The phase diagram of the system depends only on the packing fraction $\eta := N\pi\sigma^2/L^2$. In the following, the letters a, b, c, \dots , label hard-disk configurations of N disks, given by the coordinates of the disk centers \mathbf{r}_i . The letters i, j, k number disks.

2.1. Balance conditions

Markov-chain Monte Carlo algorithms are governed by balance conditions for the flows $P_{a \rightarrow b} := \pi_a p_{a \rightarrow b}$ from configuration a to configuration b (see Fig. 1); $p_{a \rightarrow b}$ is the conditional probability to move from a to b , given that the system is in a . To converge towards the stationary distribution π_a , the *global* balance condition must be satisfied: The total flow onto configuration a must equal the total flow out of a ,

$$\sum_c P_{c \rightarrow a} = \sum_c P_{a \rightarrow c}. \quad (1)$$

The local Monte Carlo algorithm, introduced by Metropolis et al. in 1953 [15] (see Fig. 2), uses the more restrictive *detailed* balance condition $P_{a \rightarrow b} = P_{b \rightarrow a}$ for which the net flow between each pair of configurations a and b is zero. Moving from configuration $a = \{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_N\}$ to $b = \{\mathbf{r}_1, \dots, \mathbf{r}_i + \boldsymbol{\delta}, \dots, \mathbf{r}_N\}$ involves sampling the disk i to be displaced and the displacement $\boldsymbol{\delta}$. For detailed balance, the probability to sample $\boldsymbol{\delta}$ at \mathbf{r}_i must equal the probability to sample $-\boldsymbol{\delta}$ at position \mathbf{r}'_i . In order to be ergodic, the displacements $\boldsymbol{\delta}$ are chosen such that each disk can eventually reach any position in the system.

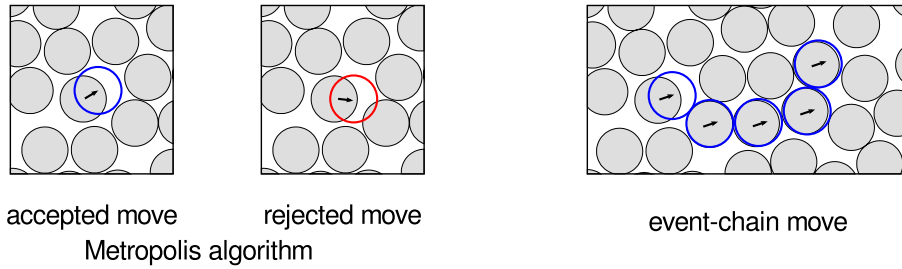


Figure 2. Monte Carlo moves for hard disks. *Left:* Accepted and rejected Metropolis moves. *Right:* Event-chain move. The sum of individual displacements equals a predefined value ℓ . With periodic boundary conditions, the event-chain move is rejection-free.

Unlike the local Monte Carlo algorithm, a single move of the event-chain algorithm [19] may displace several disks. An event-chain move is parametrized by a total displacement ℓ and a direction \mathbf{e}_ℓ , which together form a vector $\boldsymbol{\ell} := \ell \mathbf{e}_\ell$. The move starts by sampling a disk i and “sliding” it in the \mathbf{e}_ℓ direction until it hits another disk j , or at most for the distance ℓ . The disk j is then displaced in its turn, also in the \mathbf{e}_ℓ direction, see Fig. 2. This process continues until the displacements of the individual disks sum up to ℓ . After this, a new disk and possibly a new direction are sampled for the next move. With periodic boundary conditions, no rejections occur in this algorithm. For a given displacement vector $\boldsymbol{\ell}$, any disk configuration a can reach N other configurations, using each of the N disks to start an event chain. Likewise, a can be reached from N other configurations which may be reconstructed by event chains with displacement vector $-\boldsymbol{\ell}$. This implies that the event-chain satisfies the global balance condition, Eq. (1). If the vectors $\pm\boldsymbol{\ell}$ are equally likely, it also satisfies detailed balance. In order to be ergodic, the displacements $\boldsymbol{\ell}$ must span space: By choosing $\mathbf{e}_\ell \in \{\mathbf{e}_x, \mathbf{e}_y\}$, the event-chain algorithm realizes the maximal global balance (see Fig. 1), where flow between two configurations is possible only in one direction. This version is more efficient than detailed balance versions (for example, $\pm\mathbf{e}_x$ and $\pm\mathbf{e}_y$) [19]. It is again possible to alternate repeated moves in the \mathbf{e}_x direction with repeated moves in \mathbf{e}_y without destroying the correctness of the algorithm. For displacements ℓ smaller than the mean free path l_{mfp} , the event-chain algorithm is roughly equivalent to the local Monte Carlo algorithm. It accelerates for increasing ℓ , and for ℓ much larger than the mean free path, it is about two orders of magnitude faster than the local Monte Carlo method, and about ten times faster than the best current implementations [16] of event-driven molecular dynamics (see Ref. [25]).

2.2. Correlation times and orientational order

The characteristic challenge of numerical simulations for the hard-disk model resides in the extremely long correlation time. This is illustrated in Fig. 3 using snapshots of configurations obtained during a long simulation run. The system is quite small and not extremely dense, yet

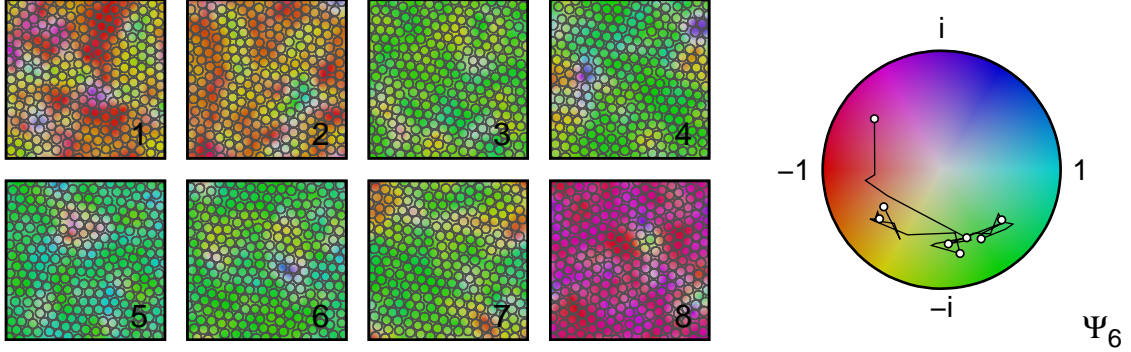


Figure 3. Local Monte Carlo evolution of 16^2 disks in a square box with periodic boundary conditions at packing fraction $\eta = 0.707$. *Left:* Disk configurations and their local orientational field ψ_j for one simulation run. Frame are separated by $100,000N$ iterations (10^5 sweeps) of local Monte Carlo. The slow decorrelation of the orientation is manifest. *Right:* Evolution of the global orientational order parameter Ψ_6 , Eq. (3), in the complex plane, for the same simulation run.

correlations in the orientation of the system persist over millions of Monte Carlo moves. To quantify the orientations and their correlations, we consider the local orientational field

$$\psi_j := \sum_{k=1}^{N_j} w_{j,k} \exp(6i\phi_{j,k}), \quad (2)$$

where N_j is the number of Voronoi neighbors of disk j . The $w_{j,k}$ (with $\sum_k w_{j,k} = 1$) are normalized weights according to the length of the Voronoi interface between disks j and k , and $\phi_{j,k}$ is the angle of the vector between the disk centers [26]. The average of Eq. (2) over all disks yields the global orientational order parameter,

$$\Psi_6 := \frac{1}{N} \sum_j \psi_j. \quad (3)$$

In a square box, the mean value of Ψ_6 is zero because of the $\phi_{j,k} \rightarrow \phi_{j,k} + \pi$ symmetry, and its correlation function

$$C_6(\Delta t) := \frac{\langle \Psi_6(t) \Psi_6^*(t + \Delta t) \rangle_t}{\langle |\Psi_6(t)|^2 \rangle_t}. \quad (4)$$

decays to zero for infinite times Δt . We conjecture that Ψ_6 is the slowest observable in the system. For large times, global orientational correlations decay exponentially, $C_6(\Delta t) \propto \exp(-\Delta t/\tau)$, and we obtain the empirical correlation time τ from an exponential fit to C_6 .

3. Polytope representation of event-chain moves

Event-chain moves along a single direction $\mathbf{e}_\ell = \ell/\ell$ sample a restricted configuration space. For the remainder of this section, we take the chains to move in the positive x direction, unless specified otherwise, to simplify the notation. Since all y coordinates are fixed, two disks whose y coordinates differ by less than 2 radii cannot slide across each other, and their relative order is fixed. Furthermore, while in x collision mode, any disk can collide with not more than six

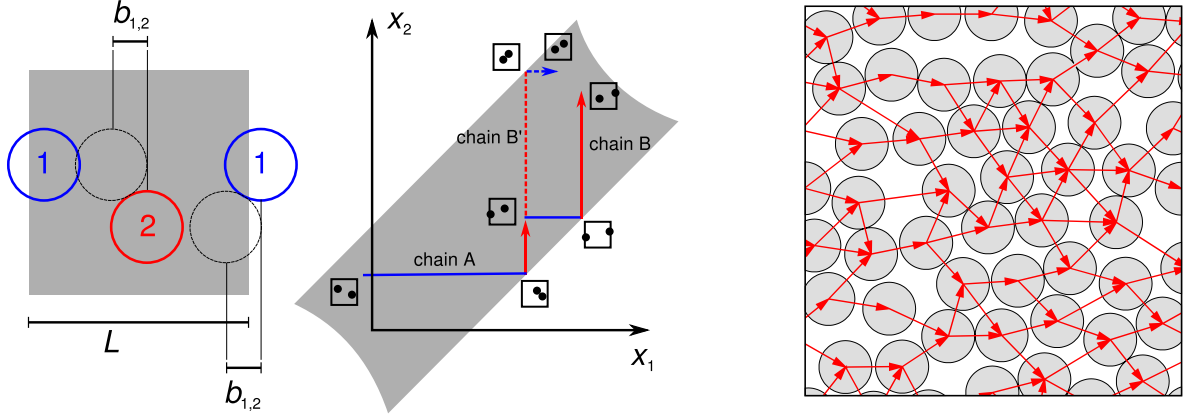


Figure 4. Event-chain move and polytope representation. *Left:* Two disks in a periodic box. The constraints of Eq. (5) are $x_1 \leq x_2 - b_{1,2}$ and $x_2 \leq x_1 + L - b_{1,2} = x_1 - b_{2,1}$. *Center:* Molecular dynamics evolution in the polytope corresponding to two event chains with moves of disk 1 (blue segments) and disk 2 (red segments). The trajectory begins with disk 1, and it depends on the choice of the starting disk (1 or 2) for the second chain. Periodic boundary conditions are ignored for clarity. Snapshots of the configuration are sketched along the trajectory. *Right:* Hard-disk configuration with its constraint graph for motion along the x axis. Each node has at most three forward and three backward links. This graph is invariant under event-chain moves in the x direction.

other disks, at most three in the forward direction, and at most three in backward direction (see Fig. 4). The collision partners of a disk may include itself, because of boundary conditions. The relations among disks constitute a *constraint graph*, which expresses the partial order between them (see Fig. 4). This graph remains invariant while performing event-chain moves in the $\pm \mathbf{e}_x$ direction. Each directed edge from i to k corresponds to a linear inequality for the x coordinates of the disks i and k :

$$x_i \leq x_k - b_{i,k}, \quad (5)$$

with $b_{i,k} := \sqrt{4 - (y_i - y_k)^2}$. The constant $b_{i,k}$ can be adjusted to also account for periodic boundary conditions in the x direction. The inequalities Eq. (5) imply that no more than three forward collision partners can be present¹.

The system of linear inequalities Eq. (5) delimit a subset of the N -dimensional space of x coordinates $X = (x_1, \dots, x_N)$, an N -dimensional polytope, bounded by at most $3N$ hyperplanes. This convex object is easier to analyze than the highly intricate $2N$ -dimensional configuration space of the full hard-disk problem. The polytope is unbounded in the $(1, 1, \dots, 1)$ direction in consequence of the periodic boundary conditions, since uniform translation of all the disks is always permitted. Also, since the constraint graph is invariant under event-chain moves in the x direction, so is the polytope. However, the polytope becomes bounded by taking a section orthogonal to $(1, 1, \dots, 1)$.

¹ A superset of the actual constraint graph can be computed from efficient local criteria. This superset contains redundant inequalities, but describes the same polytope; it is, in particular, useful for the practical implementation: If i collides forward with j , and j collides forward with k , we have $x_i \leq x_k - b_{i,j} - b_{j,k}$; if now $b_{i,j} + b_{j,k} > b_{i,k}$, the disks i and k can never come into contact; disk j covers disk k . Applying this rule iteratively, the disks in the forward direction can be reduced to at most three: at most one each with $y \in (y_i - 1, y_i + 1)$, with $y \in [y_i + 1, y_i + 2)$ and with $y \in (y_i - 2, y_i - 1]$. Thus, each node in the constraint graph has at most degree six.

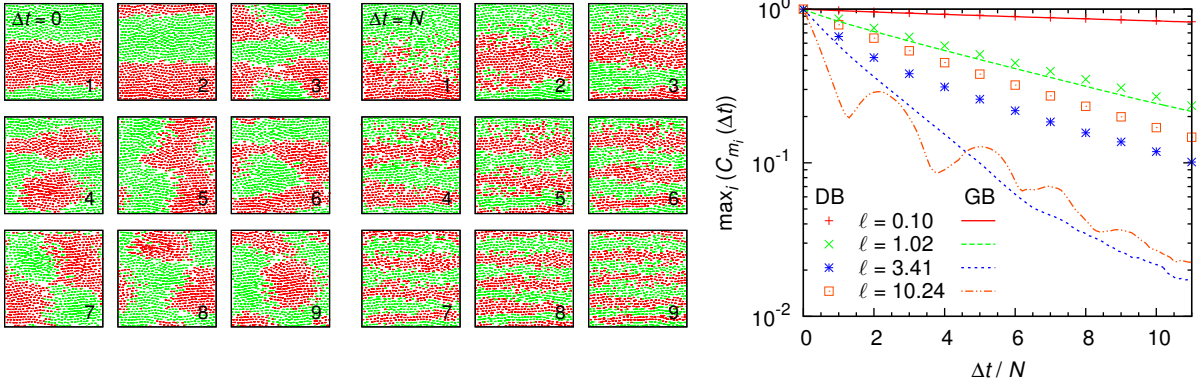


Figure 5. Relaxation dynamics in the invariant polytope, for a given initial configuration of the 32^2 hard-disk system at packing fraction $\eta = 0.698$. *Left:* Slowly decaying modes. Configurations are shown with red disks moving in the $+x$ direction and green disks in $-x$. The modes shown are the eigenvectors of the largest eigenvalues of $U(0)$. *Center:* Remaining correlation after $\Delta t = N$ of event-chain moves in the horizontal direction. These are the largest eigenvectors of $U(N)$. Correlations in the horizontal direction have all but disappeared. *Right:* Decay of the slowest modes, for the event-chain simulations with various total displacements ℓ , and for both the global (GB) and the detailed (DB) balance version.

In the invariant polytope, an event chain of total displacement ℓ corresponds to a molecular dynamics evolution of duration ℓ : Displacing the i -th disk corresponds to the “particle” X moving in the i -th coordinate direction, and each collision event (the transfer of momentum from one disk to another) to a right-angle reflection at the facets of the polytope (see Fig. 4). The construction of the event-chain move is finished at time ℓ . The next move involves the sampling of a new starting disk and possibly of one of the $\pm \mathbf{e}_\ell$ directions. In the invariant polytope, this is the choice of new velocities. Local Monte Carlo on the other hand, if restricted to moves in x direction, implements diffusive motion in the invariant polytope².

The invariant constraint graph allows for fast lookup of possible collision partners, and may even replace the customary cell grids (see, for example, Section 2.4 of Ref. [28]). While computation of the actual constraint graph requires depth search, a superset sufficient for practical computations can be computed efficiently, see the footnote on page 5.

3.1. Correlation functions in the invariant polytope

Although the sampling problem from the invariant polytope concerns a convex body, it is notoriously nontrivial [27]. The inequalities Eq. (5) essentially amount to a system of coupled one-dimensional hard-disk problems. To study the relaxation behavior effected by the event-chain algorithm in the polytope, we consider the cross-covariance of the disk coordinates,

$$U_{ij}(\Delta t) := \langle \tilde{x}_i(t + \Delta t) \cdot \tilde{x}_j(t) \rangle_t, \quad (6)$$

where $\tilde{x}_i(t)$ is the x coordinate of the disk i , compensated for the overall translation of the system due to the event-chain moves,

$$\tilde{x}_i(t) := x_i(t) - \frac{\alpha t}{N} - \left\langle x_i(t) - \frac{\alpha t}{N} \right\rangle_t. \quad (7)$$

² To preserve the polytope, a “sliding” version of local Monte Carlo must be considered: The move $\mathbf{r}_i \rightarrow \mathbf{r}_i + \delta$ is valid only if all intermediate positions $\mathbf{r}_i + \alpha \delta$ with $\alpha \in [0, 1]$ yield legal hard-sphere configurations.

Here, α is 1 for the global balance version of the event-chain algorithm (chains only in $+x$ direction), and 0 for the detailed balance version ($\pm x$). The eigenvectors of $U(0)$ are the polytope's normal modes m_i , $i = 1, \dots, N$, in the sense of principal component analysis. The nature of the modes m_i depends on the structure of the invariant polytope and captures the relative order of colliding disks and their frozen-in y coordinates. The normal modes to the largest eigenvalues are large-scale cooperative rearrangements of the disks (see Fig. 5). They are the slowest modes to decay under both local and event-chain Monte Carlo and govern the global decorrelation of the disk configuration. In particular, two modes dominated by antiparallel flow bands are very slow to decay (mode 1 and 2 in Fig. 5).

At delay times $\Delta t > 0$, the cross-covariance $U_{ij}(\Delta t)$ captures residual correlations among the disk coordinates. The event-chain moves couple more efficiently to the longitudinal modes of the system, and we find that after $\Delta t \approx N$, the event-chain algorithm has virtually erased longitudinal correlations. The most prominent residual correlations carry a transverse band structure (see Fig. 5). The result is a substantial decrease in efficiency of the algorithm for simulated duration in a single direction larger than $\approx N$.

To estimate the convergence time, we study the projection of the system's evolution $X(t)$ onto a single mode, $X(t) \cdot m_i$. The autocorrelation function

$$C_{m_i}(\Delta t) := \frac{\langle (X(t) \cdot m_i)(X(t + \Delta t) \cdot m_i) \rangle_t}{\langle (X(t) \cdot m_i)^2 \rangle_t} \quad (8)$$

is, for short chain lengths ℓ , monotonously decaying. Larger chain lengths accelerate the decay, as the coupling to large-scale modes is improved (Fig. 5). For chains spanning several times the box, however, the autocorrelation functions C_{m_i} develop oscillations with very weak damping, offsetting the benefits of longer chains. The detailed balance version of event-chain Monte Carlo is generally slower and less prone to oscillations. For optimal performance, the global balance version should thus be used with ℓ larger, but on the order of $l_{\text{mfp}}\sqrt{N}$, and for times $\theta \approx N$ (see Fig. 5). For disk configurations larger than the correlation length, ℓ can be reduced appropriately.

3.2. Convergence of the full hard-disk problem

The invariant polytope representation allows us to interpret the convergence of the full hard disk sampling problem. The conceptually simplest Monte Carlo algorithm for hard disks consists entirely in polytope sampling: One iteration amounts to direct sampling a new configuration a_{n+1} from the invariant polytope of the starting configuration a_n , and exchanging the x and y coordinates of all the disks. This Markov-chain algorithm satisfies detailed balance. In our experiments, the timescale τ , measured in iterations, for relaxation to equilibrium increases only as $N^{1/4}$ for large systems, implying that most of the complexity of the hard-disk sampling problem resides in the polytope sampling.

Since direct sampling is a hard problem for high-dimensional polytopes (see Section 3.3), we replace it by Markov chains of a fixed number of event-chain moves, in effect performing molecular dynamics in the invariant polytopes for fixed duration θ :

$$\left\{ \begin{array}{l} \text{polytope MD} \\ \text{in } x \text{ direction} \\ \text{for duration } \theta \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \text{polytope MD} \\ \text{in } y \text{ direction} \\ \text{for duration } \theta \end{array} \right\} \rightarrow \dots \quad (9)$$

This algorithm satisfies detailed or global balance depending on the version of the event-chain algorithm that is used for polytope sampling.

We study the influence of the switching interval θ on convergence properties. In Fig. 6, the autocorrelation function C_6 of the complex order parameter Ψ_6 is plotted vs. cumulative

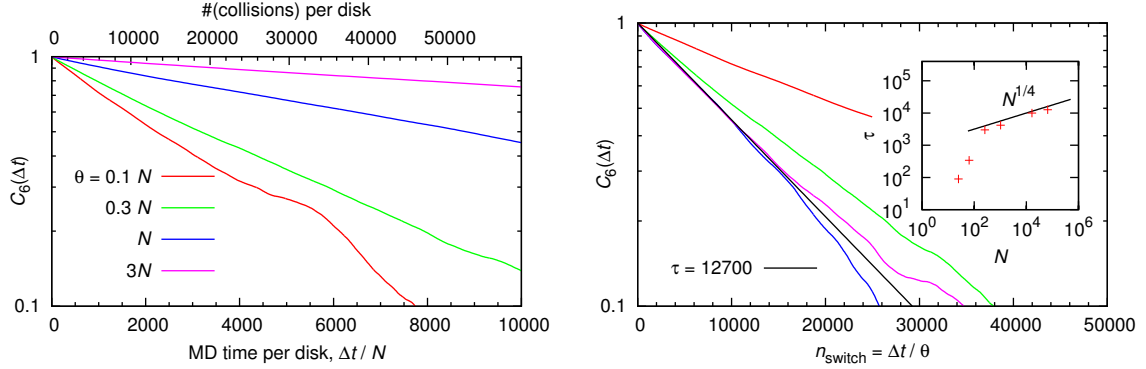


Figure 6. *Left:* Decay of the Ψ_6 autocorrelation function $C_6(\Delta t)$ for several switching intervals θ , as a function of the simulated MD time (*bottom axis*), or alternatively, the number of collisions per disk (*top axis*). *Right:* Decay of C_6 as a function of the number of x/y switches n_{switch} . As θ approaches N , the curves approach the limit of direct sampling from the polytope, with a mixing time of τ cycles. All curves were averaged from systems of $N = 256^2$ disks at packing fraction $\eta = 0.698$; the chain length was $\ell = 6.5 \cdot 10^3$. *Inset:* The mixing time τ first increases rapidly with system size, but only grows as $N^{1/4}$ for larger systems (also $\eta = 0.698$).

molecular dynamics time. C_6 decays most quickly when the switching interval θ is small, but the decay speed deteriorates very slowly with θ . Only at $\theta \approx N$ (corresponding to about 6-7 collisions per disk at these densities), the algorithm becomes notably less efficient. The efficiency drop thus follows the decay of longitudinal (in x direction) correlations in the invariant polytope, and is to be expected from the results in Section 3.

In the limit $\theta \rightarrow \infty$, the event-chain algorithm realizes direct sampling in the invariant polytope. The approach to this limit is illustrated in Fig. 6 by plotting C_6 against the number of x/y switching cycles n_{switch} . As the switching interval θ increases, the autocorrelation functions approach an asymptotic curve $\propto \exp(-n_{\text{switch}}/\tau)$, where τ is the correlation time of the direct sampling algorithm. We find that for practical purposes, event-chain Monte Carlo reaches the asymptotic regime for $\theta \approx N$, and thus samples an approximately independent point in the invariant polytope in $O(N)$ operations. Importantly, the correlation time $\tau(N)$ increases rapidly only for small system size N . After the system size surpasses the correlation length, τ grows only as $N^{1/4}$.

3.3. Application to general polytopes

The invariant polytope is bounded by hyperplanes which are normal to $N - 2$ coordinate axes and have unit derivative along the remaining axes. By choice of the ℓ , the molecular dynamics evolution is aligned with the coordinate axes at all times, and computations of intersections are of complexity $O(1)$. As shown in Section 3.2, the event-chain algorithm seems to achieve an effective mixing time of $O(N)$ collision events, so that the cost of sampling the hard-disk polytope appears as $O(N)$.

The event-chain algorithm also allows to sample general polytopes. Direct sampling from polytopes is straightforward only in low dimensions N , especially in $N = 2$: A two-dimensional polytope with n edges (a convex n -sided polygon), can be decomposed into n triangles, using an interior point. Triangles may then be sampled according to their areas, and a random point may be sampled inside the sampled triangle (see, e.g. chap. 6.2 of [28]). In higher dimensions N , triangulation by simplices generalizes this decomposition. Since polytopes such as the invariant hard-disk polytope have an exponential number of facets, direct sampling algorithms are no

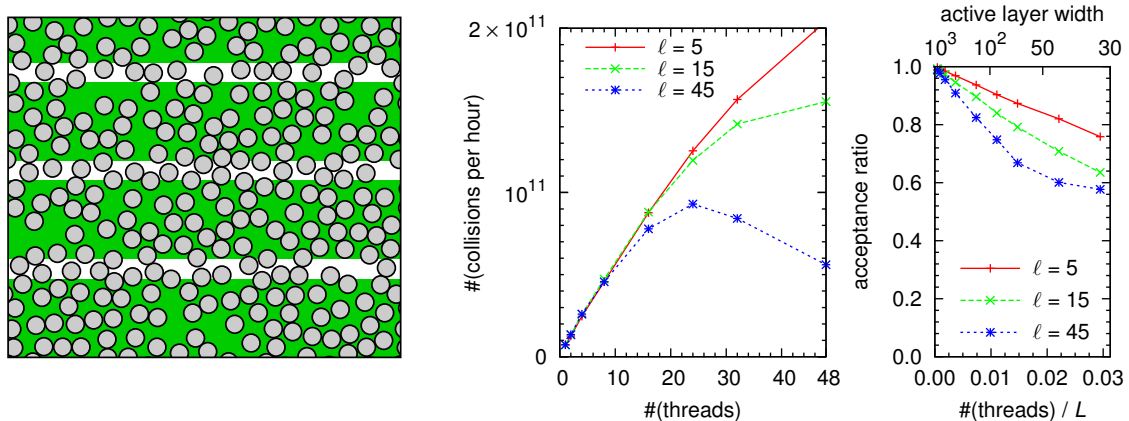


Figure 7. *Left:* Two-color stripe scheme with active (green backdrop) and isolation layers for the event-chain algorithm. Chains may run simultaneously if they are located in different stripes. *Center:* Scaling of the isolation layer algorithm on a shared-memory machine (Opteron 6276, 2.3 GHz), for a $N = 4096^2$ disk packing at $\eta = 0.698$. We plot the number of collisions in accepted chains per hour of computation. On the same machine, the serial version has a performance of about $8.5 \cdot 10^9$ collisions per hour. The event-chain routine is the same in both programs. *Right:* The acceptance ratio for chains depends on the thickness of the active layers (which decreases as more threads are added) and the total displacement ℓ of the chains.

longer practical. Markov-chain sampling [29, 30, 31, 32] achieves mixing times of $O(MN)$ steps, where M is the number of bounding hyperplanes ($M \leq 3N$ for hard disks), and where each move may be implemented in $O(M)$ steps. It will be interesting to see how event-chain polytope sampling compares with existing polytope sampling methods, in particular the ‘hit-and-run’ algorithms.

4. Parallel Monte Carlo algorithms for hard disks

In view of the long running times of Monte Carlo simulations and of the current standstill in computer clock speeds, it is essential to develop *parallel Monte Carlo methods* which distribute the work load among several *threads* performing independent computation with as few communication as possible. Such methods will allow to study not only the standard hard disk ensemble, but also related systems such as soft disks and polydisperse disk packings. However, parallel Monte Carlo algorithms for continuum systems pose many more problems than for lattice models, for example the Ising spins, where straightforward parallel application of local Metropolis updates converges to the Boltzmann distribution [33].

4.1. Parallel implementation of local Monte Carlo

A massively parallel implementation of the local Monte Carlo algorithm was applied recently to the hard-disk melting problem [34, 25]. It sets up square cells according to a four-color checkerboard pattern. Disks in same-color cells can be updated simultaneously, but moves across cell boundaries are rejected. To ensure ergodicity, a new cell grid must be sampled periodically. Massive parallelism of ~ 1500 threads on a graphics card offsets the slowness of local Monte Carlo compared to event-chain algorithm Monte Carlo [25]. These calculations confirmed the first-order liquid-hexatic phase transition in hard disks [8].

4.2. Parallel implementations of event-chain Monte Carlo

For parallel implementations of event-chain Monte Carlo, we consider only parallel threads that run chains in the same direction $\pm \mathbf{e}_\ell$. This minimizes the chance that two chains cross each other and move the same disks. It also allows us to apply the invariant polytope framework of Section 3. It is instructive to realize that the effects of an event-chain move can be summarized in the difference vector of the new and old x coordinates: $\Delta X = X(\ell) - X(0)$, with $X(t) = (x_1(t), \dots, x_N(t))$. Moreover, if two chains are *independent*, meaning their sets of disks touched are disjoint, the net effect of running both chains is the sum of their individual difference vectors, $\Delta X^{\text{net}} = \Delta X^{(1)} + \Delta X^{(2)}$. If, however, any disk is touched by both chains, the chain reaching this disk earlier in MD time has precedence, the later chain sees a modified environment, and consequently takes a different evolution. Thus, interdependent chains cannot be added arithmetically³. The primary obstacle in parallelizing event-chain Monte Carlo consists in preserving the correct causal relations between subsequent chains, as required for the convergence to correct equilibrium distribution.

In the following, we discuss three strategies to parallelize event-chain Monte Carlo. The *predict/execute algorithm* distributes work among threads for a model of chains that follow each other chronologically. The effects ΔX of several chains are predicted in advance from the current disk configuration. The effects of the chains are then applied to the system state in the chronological order in which the starting disks were sampled. To detect conflicts, it is sufficient to compute the intersection of the set of disks touched by the current chain and of the chains that ran since the beginning of planning; if this intersection is not empty, the chain has to be recomputed from the updated state of the disk configuration. Planning and execution of chains can proceed in parallel on a shared-memory machine. Using lock-free data structures, we attain collision rates in excess of 10^{11} per hour in x collision mode on a four-processor machine. Due to its serial nature, this algorithm does not scale well beyond a few threads, however; with too many chains predicted in advance, the probability for recomputations rises. Moreover, switching between x and y collision modes requires reinitialization of the data structures and is rather expensive.

A variation of the four-color scheme adapted to the event-chain algorithm partitions the system in horizontal stripes, separated by *frozen isolation layers* of thickness ≥ 2 disk radii (see Fig. 7). Disks with their centers in the isolation layers are kept fixed, and thus guarantee the independence of chains running in neighboring stripes. To preserve the isolation layers, chains colliding with a frozen disk are rejected. As there are rejected moves, the global balance condition is no longer guaranteed: the number of accepted forward chains can be different from the number of accepted backward chains (see Section 2.1). When allowing chains in both the $\pm \mathbf{e}_\ell$ directions, however, the isolation layer algorithm satisfies detailed balance. Furthermore, in order to limit the rejection rate, the per-chain total displacement ℓ has to be kept lower than in the serial algorithm. In view of the discussion of Section 3, these necessities reduce somewhat the efficiency of the method. Due to the isolation layers, the accessible configuration space is restricted, and for ergodicity, the layer boundaries have to be resampled periodically, as in the four-color version of local Monte Carlo.

We have implemented the isolation layer algorithm in parallel on a shared-memory machine. Using several cores in parallel, it is possible to achieve effective collision rates which are 10–30 times the single-core performance (see Fig. 7), for systems of sufficient size. For systems too small, less threads can be used without shrinking the active strips to a point where the acceptance ratio becomes a limiting factor. Systems of physical interest, however, are on the order of $N = 1024^2$, and allow to use 10–20 cores with moderate ℓ . At this time, the algorithm is not bound by rejection rates, but by communication between threads.

³ Note, however, that due to the convexity of the accessible configuration space, the arithmetic average of two moves, $(\Delta X^{(1)} + \Delta X^{(2)})/2$, is always admissible.

Finally, the event-chain scheme is not fundamentally limited to a single moving disk at any time. We may indeed launch multiple *concurrent chains*, which run at the same simulated MD time, and interact with each other. This is different from the parallel simulation of chains which interact in sequential manner. In the invariant polytope picture, multiple concurrent chains correspond to choosing more general initial conditions, where more than one disk is given an initial velocity of 1. After time ℓ , multiple chains have executed, and possibly interacted with each other; there is no rejection in this algorithm. The problem has some resemblance with event-driven molecular dynamics, because the scheduling of collisions must be foreseen, but there are several simplifications: all velocities are in the same direction and of magnitude 0 or 1. As a consequence, two moving disks cannot collide with one other; however, the faithful simulation of chains close by and possibly interacting requires careful synchronization among threads. In our experiments, this limits the speedup by parallelization. Our most efficient method at this point is the isolation layer algorithm.

5. Conclusion

We have reached in this paper a better understanding of the event-chain Monte Carlo algorithm for the hard-disk sampling problem. By restricting the algorithm to chains in a single direction, a connection appears to the well-known problem of sampling random points from a polytope: A move of the event-chain algorithm consists in performing a finite-time molecular dynamics simulation in the invariant polytope of the disk configuration. This connection offers new strategies to solve the hard-disk sampling problem in terms of polytope sampling; it also suggests to investigate the utility of event-chain methods for the sampling of general polytopes. Finally, it will be interesting to study the combinatorial structure of the typical invariant polytope, and its dependence on thermodynamical parameters.

By the study of correlation functions, we have shown that the Monte Carlo relaxation process in the invariant polytope separates into two phases: A rapid longitudinal relaxation, followed by a much slower relaxation of the transverse degrees of freedom. We have given recommendations for the parameters of the algorithm based on these results. Finally, we have discussed several strategies for parallelizing Monte Carlo algorithms for hard disks, alleviating the problem of the long simulation times in hard disk Monte Carlo. The parallelization of the hard-disk ensemble remains challenging due to its unique combination of very little actual computation and long correlation times. Efficient methods to tackle the hard-disk ensemble are, however, crucial in order to treat related systems such as soft disks with the same level of success as the hard disks. New concepts such as the link to polytope sampling will be essential in this effort.

Acknowledgment

We thank P. Diaconis, E. P. Bernard, S. Leitmann and M. Hoffmann for fruitful discussions.

References

- [1] L. Boltzmann, Sitzber. Akad. Wiss. Wien, Math. Naturw. Kl. (Pt. 2a) (1896).
- [2] J. L. Lebowitz and O. Penrose, *J. Math. Phys.* **5** 841 (1964).
- [3] J. G. Kirkwood, E. Monroe, *J. Chem. Phys.* **8** 845 (1940).
- [4] S. Asakura, F. Oosawa, *J. Chem. Phys.* **22** 1255 (1954).
- [5] B. J. Alder, T. E. Wainwright, *Phys. Rev.* **127**, 359 (1962).
- [6] J. M. Kosterlitz, D. J. Thouless, *J. Phys. C: Solid State Phys.* **6** 1181 (1973); B. O. Halperin, D. R. Nelson *Phys. Rev. Lett.* **41** 121 (1978).
- [7] C. H. Mak, *Phys. Rev. E* **73** 065104(R) (2006).
- [8] E. P. Bernard, W. Krauth, *Phys. Rev. Lett.* **107**, 155704 (2011).
- [9] K. Zahn, R. Lenke, G. Maret, *Phys. Rev. Lett.* **82** 2721 (1999).
- [10] P. Diaconis, *J. Stat. Phys.* **144** 445 (2011).
- [11] Y. G. Sinai, *Russian Mathematical Surveys* **25**, 137 (1970).
- [12] N. Simanyi, *Inventiones Mathematicae* **154**, 123 (2003).

- [13] B. J. Alder, T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).
- [14] D. C. Rapaport, *J. Comp. Phys.* **34**, 184 (1980).
- [15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *J. Chem. Phys.* **21** 1087 (1953).
- [16] M. Isobe, *Int. J. Mod. Phys. C* **10**, 1281 (1999).
- [17] C. Dress, W. Krauth, *J. Phys. A, Math. Gen.* **28** L597 (1995).
- [18] A. Jaster, *Phys. Rev. E* **59**, 2594 (1999).
- [19] E. P. Bernard, W. Krauth, D. B. Wilson, *Phys. Rev. E* **80**, 056704 (2009).
- [20] R. H. Swendsen, J. S. Wang, *Phys. Rev. Lett.* **58** 86 (1987).
- [21] U. Wolff, *Phys. Rev. Lett.* **62** 361 (1989).
- [22] R. Kannan, M. W. Mahoney, R. Montenegro, in T. Ibaraki, N. Katoh, H. Ono (Eds.): Algorithms and Computation, 14th International Symposium, ISAAC 2003. Proceedings. Lecture Notes in Computer Science 2906, Springer (2003).
- [23] D. B. Wilson, *Random Struct. Algorithms* **16**, 85 (2000).
- [24] C. Chanal, W. Krauth, *Phys. Rev. E* **81** 016705 (2010).
- [25] J. A. Anderson, M. Engel, S. C. Glotzer, M. Isobe, E. P. Bernard, W. Krauth, arXiv:1211.1645.
- [26] W. Mickel, S. C. Kapfer, G. E. Schröder-Turk, K. Mecke, to appear in *J. Chem. Phys.* (2013); arXiv:1209.6180.
- [27] M. E. Dyer, A. M. Frieze, *SIAM J. Comput.* **17** 967 (1988); V. Kaibel, M. E. Pfetsch, in *Algebra, Geometry and Software Systems*, edited by M. Joswig, N. Takayama (Springer, 2003); arXiv:math/0202204.
- [28] W. Krauth, *Statistical Mechanics: Algorithms and Computations*, Oxford University Press (2006).
- [29] R. L. Smith, *Operations Research* **32**, 1296 (1984).
- [30] P. A. Rubin, *Communications in Statistics - Simulation and Computation* **13**, 375 (1984).
- [31] M. Dyer, A. Frieze, R. Kannan, *J. ACM* **38**, 1 (1991).
- [32] R. Kannan and H. Narayanan, *Mathematics of Operations Research* **37**, 1 (2012).
- [33] B. A. Berg, *Markov Chain Monte Carlo Simulations And Their Statistical Analysis*, World Scientific, (2004).
- [34] J. A. Anderson, E. Jankowski, T. L. Grubb, M. Engel, S. C. Glotzer, arXiv:1211.1646.